Space time tradeoffs for the solution of banded linear equations and other problems

J.E. Savage

IRIA

CENTRE DE ROCQUENCOURT

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél:954 90 20

Rapports de Recherche

N° 88

SPACE-TIME TRADEOFFS FOR THE SOLUTION OF BANDED LINEAR EQUATIONS AND OTHER PROBLEMS

John E. SAVAGE

Août 1981

Space-Time Tradeoffs for the Solution of Banded Linear Equations and Other Problems †

John E. Savage

Department of Computer Science
Brown University
Providence, RI 02912, USA

ABSTRACT

Upper and lower bounds are derived on the time and temporary storage space needed for a number of problems, including banded matrix multiplication, inversion and the resolution of a banded set of linear equations. Also considered are lower bounds for certain binary integer functions, such division and square roots, and for the class of transitive functions, which includes sorting and matrix inversion.

RESUME

Les bornes superieures et inferieures sont obtenues sur le temps et l'espace temporaire requis pour une ensemble de problemes, en particulier, pour la multiplication des matrices bandes, pour l'inversion de telles matrices, et pour la resolution d'un systemes d'equations lineaires avec matrice bande. Des bornes inferieures sont obtenues aussi pour certaines functions sur les entiers, comme la division et la racine carree des entiers, et pour les functions transitives, donc le tri et l'inversion des matrices sont des exemples.

August 26, 1981

† The research reported here was supported in part by INRIA, Rocquencourt, France, by the University of Paris-Sud, Orsay, France, and by the National Science Foundation under Grant ECS 80-24637.

Keywords: tradeoffs, space, time, matrix multiplication, banded matrices, matrix inversion, linear equations, stack algorithms.

1. INTRODUCTION

Space-time tradeoffs are important characterizations of the limits that exist on the simultaneous use of computational resources. They are often stated as lower bounds on the product of space and time that must hold for a given class of algorithms and a given problem. Their value lies in providing a means to show for a given problem and an algorithm for it, how close the algorithm is to being optimal for these two resources.

Grigoryev [1] has introduced a framework for the derivation of lower bounds on the product of temporary storage space S and computation time T for the class of straight-line algorithms. He has used his framework to show that the product of two p×p Boolean matrix multiplication and of two n-degree polynomial multiplication over GF(2) require $ST = \Omega(p^3)$ and $ST = \Omega(n^2)$, respectively. Tompa [2] has applied the Grigoryev approach to superconcentrators, the Discrete Fourier Transform, sorting and merging, and has obtained quadratic lower bounds in each case in the number of input variables. Swamy [3] has independently derived the same type of bound for sorting and Savage and Swamy [4] have derived quadratic lower bounds for binary integer multiplication. Ja'Ja' [5] has shown that, over a field of sufficiently large characteristic, the inverse of a p×p matrix requires $ST = \Omega(p^4)$, solving a system of p equations in p unknowns requires $ST = \Omega(p^3)$, and computing the set of elementary symmetric functions in n unknowns requires $ST = \Omega(n^2)$.

In this paper we extend somewhat the Grigoryev method to illuminate the weakest conditions under which it can be applied, which are *I/O-oblivious* † algorithms. We also apply this method to a new class of problems. These consist primarily of problems involving banded matrices, such as banded matrix multiplication, inversion and the solution of a set of banded linear equations, but also include transitive functions and certain binary integer powers and reciprocals. The class of transitive functions is shown to include the problems of sorting and matrix inversion. The lower bounds for matrix inversion thus derived apply to a much larger class of fields than the class for which Ja'Ja' [5] has derived the first results for this problem.

Banded matrices occur very often in practice in such areas as finite element analysis, the solution by finite differences of Laplace's equation, and solving for the voltages and currents in linear electrical networks [6, p.14]. In fact the author's interest in this problem was motivated by a result of Eisenstat et al [7] who show that a banded set of linear equations that are positive definite can be solved with considerably less space than is normally used. Their result is described and improved upon later.

We show that to multiply two p×p banded matrices of bandwidth b over a field F or to solve a set of linear equations that has such a matrix of coefficients requires $ST = \Omega(pb^2 \log |F|)$. However, to invert such a matrix requires $ST = \Omega(p^2b^2 \log |F|)$. We also consider algorithms for these problems and show that upper bounds of the order $ST = O(pb^2 \log |F|)$, $ST = O(pbM(b)(\log p/b)^2 \log_2 |F|)$, and $ST = O(p^2M(b)(\log p/b)^2 \log_2 |F|)$ can be achieved. Here M(b) is the minimum number of operations to do b×b matrix multiplication. Thus, the lower bound for banded matrix multiplication can be met up to a multiplicative factor, and the upper bound for the resolution of a banded set of linear equations is weak by factors of M(b)/b and $(\log p/b)^2$, where the former is at least b and the latter can be as small as 1 if p is proportional to b. This weakness in the bound for solving a banded set of equations reflects the weakness in the bounds for full matrices. The upper bound for

† A class of algorithms is *I/O-oblivious* if the sequence of input reads and outputs writes is data-independent.

matrix inversion is weak by the factors of $M(b)/b^2$ and $(\log p/b)^2$, where the first may be reducible to a constant as our knowledge of matrix multiplication increases, while the second reflects the recursive nature of our algorithms.

The paper is divided into five sections. Section 2 presents a generalization of the Grigoryev lower bound argument, while Section 3 applies it to the shifting function and to a large of functions defined as powers and reciprocals of binary integers. In Section 3 we also consider the class of transitive functions which includes sorting and matrix inversion. In the Section 4 we treat banded matrix problems. Conclusions are drawn in the fifth and last section.

2. COMPUTATIONAL INEQUALITIES

The Grigoryev method provides a lower bound to the product of time and the number of temporary storage locations to compute a multi-output function from any straight-line program. In the following we generalize the result somewhat by using only those conditions which are essential to its derivation.

Definition 1: Let $M = (Q,U,V,\delta,\lambda,q_0)$ be a *finite state automaton* (FSA) with state set Q, initial state q_0, input and output alphabets U,V, transition function $\delta:Q\times U \to Q$, and output function $\lambda:M \to V$. If $q \in Q$ and $u \in U$, then $\delta(q,u) \in Q$, $\lambda(q) \in V$. Let $u_1,u_2,\ldots,u_p$ be a sequence of inputs read by M in the first through p *th* cycles. Then, the succession of states entered is $q_0,q_1,\ldots,q_p$ where $q_i = \delta(q_{i-1},u_i)$. Outputs produced are $v_0,v_1,\ldots,v_p$ where $v_i = \lambda(q_i)$. Let $f:X^n \to Y^m$ be a function with input and output variables $x_1,\ldots,x_n$ and $y_1,\ldots,y_m$ where $f(x_1,\ldots,x_n) = (y_1,\ldots,y_m)$. Then M computes f in an *I/O-oblivious* manner if q_0 is independent of f, $U = A\times D$, $X \subseteq A$, $Y \subseteq V$, and there exist *schedules* $\rho:\{1,2,\ldots,p\} \to (\{x_1,\ldots,x_n\}\cup A)\times D$, $\sigma:\{1,2,\ldots,p\} \to \{y_1,\ldots,y_m\} \cup V$ such that $u_i = \rho(i)$, $v_i = \sigma(i)$ for $1 \le i \le p$ and $|\sigma^{-1}\{y_1,\ldots,y_m\}| = m$. The machine M uses *storage* $S = \log_2|Q|$ and *I/O time* T (henceforth abbreviated *time)* defined as the number of cycles in which either $\rho(i) \in \{x_1,\ldots,x_n\}\times D$ or $\sigma(i) \in \{y_1,\ldots,y_m\}$ or both.

Thus, and FSA M computes f in an I/O oblivious manner if the time instants at which inputs and outputs of f are read and produced are data-independent. Inputs may be read multiple times, outputs are each generated once, the set of machine instructions is unrestricted except by the bounded number of states, and the number of cycles executed can be data-independent. The computational model also permits f to be computed from a straight-line program with a bounded number of registers since the second component of each input to M can be interpreted as instructions in such a program. Thus, if M computes f from such a program, the program need not be viewed as being recorded in its state. The above generalization of the computational model for which the Grigoryev method applies is motivated by an observation of Valiant [8] that the Grigoryev conditions apply to a broader class of algorithms than just straight-line algorithms.

The following is a slight generalization of the Grigoryev condition on a multi-output function in terms of which a computational inequality will be derived.

Definition 2: A function $f:X^n \to Y^m$ with input and output variables $I = \{x_1,\ldots,x_n\}$; $J = \{y_1,\ldots,y_m\}$, respectively, is (α, l, d) - *independent* for $\alpha \ge \log_2|X|$ if the following conditions hold:

(1) There exist sets $I_0 \subseteq I$, $|I_0| = c$, $J_0 \subseteq J$, $|J_0| = d$, such that

(2) for all $k \leq l$,

(3) for all sets of $k \leq c$ indices $i_1, i_2, \ldots, i_k$,

(4) for all sets of $l-k \leq d$ indices $j_1, j_2, \ldots, j_{l-k}$,

such that $x_{i_1}, \ldots, x_{i_k} \in I_0$, $y_{j_1}, \ldots, y_{j_{l-k}} \in J_0$ there is an assignment $\Theta: \{x_{i_1}, \ldots, x_{i_k}\} \cup (I-I_0) \rightarrow X$ such that the function $(y_{j_1}, \ldots, y_{j_{l-k}})$ in the remaining input variables contains at least $2^{(l-k)/\alpha)}$ points in the image of its domain.

Grigoryev's original definition was stated for $c = n$, $d = m$, $|X| = |Y| = 2$ and $\alpha = 1$. Note that any function that is (α, l, d) - independent is also (α, l', d) - independent for $l' \leq l$. Note also that $l \leq \min(c,d)$. The proof of the following result parallels that given by Grigoryev.

Theorem 1: Let $f: X^n \rightarrow Y^m$ be (α, l, d) - independent and let it be computed by an FSA M with space S and I/O-time T. Then, S and T must satisfy

$$\lceil \alpha(S+1) \rceil T \geq \frac{3}{8} dl$$

Proof

At most one input and/or output variable of f is read or produced by M per cycle. Assume without loss of generality that output variables are produced in the order $y_1, y_2, \ldots, y_m$. Consider the set of consecutive cycles beginning with the production of y_a and ending with the production of y_b, $a \leq b$. Let $x_{i_1}, \ldots, x_{i_k}$ be the input variables read during these cycles. Then, the state q of M in effect prior to the cycle in which y_r is produced and the values of $x_{i_1}, \ldots, x_{i_k}$ read determines the values of the $t = b-a+1$ output variables $(y_a, y_{a+1}, \ldots, y_b)$.

Suppose that $k \leq l-t$ for $t = \lceil \alpha(S+1) \rceil$. Then, since f is (α, l, d) - independent, it follows that for some assignment to $x_{i_1}, \ldots, x_{i_k}$ the function $(y_a, y_{a+1}, \ldots, y_b)$ has at least $2^{(t/\alpha)}$ points in the image of its domain, which exceeds $2^S = |Q|$, the number of states of M. But given an assignment to $x_{i_1}, \ldots, x_{i_k}$, it is just the state of M that determines $(y_a, y_{a+1}, \ldots, y_b)$. This contradiction implies that $k \geq l-t+1$. Since there are at least $\lfloor d/t \rfloor$ blocks of t outputs in the set J_0 of outputs for which the independence condition holds, it follows that the number of cycles in which inputs of f are read, T_I, must satisfy

$$T_I \geq \lfloor d/t \rfloor (l-t+1)$$

Also, the number of cycles in which outputs are produced, $T_0 = d$, so that the I/O-time T satisfies $T \geq \max(T_I, T_0)$. Hence,

$$T \geq \max(\lfloor d/t \rfloor (l-t+1), d)$$

from which it is straightforward to derive the desired result using the inequality $\lfloor d/t \rfloor \geq (d-t+1)/t$ and by considering the two cases $t \leq 3dl/8$ and $t \geq 3dl/8$.

If the class of algorithms considered uses only operations of the type $h: X^c \rightarrow X$ to compute functions $f: X^n \rightarrow Y^m$, then the above inequality can also be derived with $\alpha = \beta / \log_2 |X|$, $\beta \geq 1$, and S interpreted as the number of temporary storage locations or registers.

3. SPACE-TIME BOUNDS FOR FUNCTIONS

In this section the above theorem is applied to a number of previously unexamined problems and to matrix inversion. In the latter case we generalize a result of Ja'Ja' [5] which applies to matrix inversion over fields of sufficiently large characteristic. We begin with a few definitions.

Definition 3: A function f is a *subfunction* of a function h if it is obtained by an assignment to some of the input variables and/or by the suppression of some output variables.

Definition 4: The Boolean *shifting function* $f_s^{(n)}(x_0, \ldots, x_{n-1}, s_1, \ldots, s_k) = (y_0, \ldots, y_{2n-2})$ with control variables $s_1, \ldots, s_k$ realizes the mappings

$$y_j = \begin{cases} x_{j-t} & t \le j \le t+n-1 \\ 0 & \text{otherwise} \end{cases}$$

for each $0 \le t \le n-1$ by some assignment to $(s_1, \ldots, s_k)$. The Boolean *cyclic shift function* $f_{sc}^{(n)}(x_1, \ldots, x_n, s_1, \ldots, s_k) = (y_1, \ldots, y_n)$ realizes each cyclic shift of $x_1, \ldots, x_n$ for some assignment to $s_1, \ldots, s_k$. The function $h_G(x_1, \ldots, x_n, s_1, \ldots, s_k) = (y_1, \ldots, y_n)$, $x_i, y_j \in X$, $s_i \in Z$ is *transitive* of *degree n* over X if for each permutation g in the permutation group G over {1,2,...,n},

(1) there is an assignment to $s_1, \ldots, s_k$ such that $y_i = x_{g(i)}$, $1 \le i \le n$, and

(2) for each $1 \le i,j \le n$ there exists $g \in G$ such that $g(i) = j$.

A function is also *transitive* of order n if it has a subfunction of this type.

The class of transitive functions has been defined by Vuillemin [9] for the case of $|X| = 2$.

Our first result concerns the shifting function.

Proposition 1: The shifting function $f_s^{(n)}$ is $(\alpha, n(1-1/\alpha), 2n-1)$ - independent for $\alpha > 1$.

Proof

In Definition 2 let $I_0 = \{x_0, \ldots, x_{n-1}\}$, $J_0 = \{y_0, \ldots, y_{2n-2}\}$. Define $\{a_i\}$ and $\{b_j\}$ by

$$a_i = \begin{cases} 1 & i \in \{i_1, \ldots, i_k\} \\ 0 & \text{otherwise} \end{cases}$$

$$b_j = \begin{cases} 1 & j \in \{j_1, \ldots, j_{l-k}\} \\ 0 & \text{otherwise} \end{cases}$$

Consider a shift of t places. The expression

$$m_t = \sum_{i=1}^{n} a_i b_{t+i}$$

measures the intersection of selected inputs with the selected outputs that correspond to the indicated shift. Since

$$\sum_{t=0}^{n-1} m_t \le k(l-k)$$

there exists a value of t, say t_0, such that

$$m_{t_0} \le k(l-k)/n \le (l-k)l/n$$

Thus, for this shift at least

$$(l-k) - m_{t_0} \ge (l-k)(1-l/n) = (l-k)/\alpha$$

of the l-k selected outputs are in correspondence with non-selected or free inputs, where $l = n(1-1/\alpha)$. Thus, $f_s^{(n)}$ is $(\alpha, n(1-1/\alpha), 2n\text{-}1)$ - independent for $\alpha > 1$.

It follows from Definition 4 that if f is a subfunction of h and f is $(\alpha, n(1-1/\alpha), 2n\text{-}1)$ - independent, then h is also $(\alpha, n(1-1/\alpha), 2n\text{-}1)$ - independent. Since the *binary integer multiplication function* $f_m^{(n)}:\{0,1\}^{2n} \to \{0,1\}^{2n}$ contains $f_s^{(n)}$ as a subfunction, by assigning one integer to be a power of 2, we have the following immediate corollary. It provides a lower bound that is about a factor of 8 better than that derived in [4] when $\alpha = 2$.

Corollary 1.1: The binary integer multiplication function $f_m^{(n)}$ is $(\alpha, n(1\text{-}1/\alpha), 2n-1)$ - independent for $\alpha > 1$.

In [10] we show that certain powers of reciprocals and of integers represented in binary contain the shifting function as a subfunction from which we have the following corollary:

Corollary 1.2: The functions $f_R^{(n,e)}$ and $f_p^{(n,e)}$ defined by

$$f_R^{(n,e)} = \left\lfloor (2^n/y)^e \right\rfloor, \quad f_p^{(n,e)} = \left\lfloor 2^{2n} x^e \right\rfloor$$

in which $1 \le x,y \le 2^n-1$, $e = q/2^k > 0$, k and q are integers independent of n, and $e > 1$ for $f_p^{(n,e)}$, with all integers represented in binary are $(\alpha, \Theta(n(1-1/\alpha)), \Theta(n))$ - independent† for $\alpha > 1$.

We consider next the class of transitive functions.

Proposition 2: Every transitive function of order n over X is $(\alpha/\log_2|X|, n(1-1/\alpha), n)$ - independent for $\alpha > 1$.

Proof

Consider a transitive function f associated with the group G. Vuillemin notes that the sets

$$G_{ij} = \{g \in G \mid g(i) = j\}$$

for $1 \le i,j \le n$ all have size $|G_{ij}| = |G|/n$. In Definition 2 let $I_0 = \{x_1, \ldots, x_n\}$, $J_0 = \{y_1, \ldots, y_n\}$. Select some k variables in I_0 and some $(l\text{-}k)$ variables in J_0. Let $a_i, b_j \in \{0,1\}$ have value 1 if and only if x_i or y_j is selected, respectively. Then, for permutation g,

$$m_g = \sum_{i=1}^{n} a_{g(i)} b_i$$

matches occur between selected inputs and outputs. Since

$$\sum_{g \in G} = \sum_{i=1}^{n} b_i \sum_{g} a_{g(i)} \le (l-k)k|G|/n$$

† Two functions $p,q:N \to N$ are in relation $p = \Theta(q)$ if there exist constants N_0, c_1, c_2 such that for $n \ge N_0$, $c_1 q(n) \le p(n) \le c_2 q(n)$.

it follows that g_0 exists such that

$$m_{g_0} \leq (l-k)k/n \leq (l-k)l/n$$

Consequently, a correspondence can be established between $(l-k)(1-l/n)$ non-selected input variables and a like number of selected output variables. Thus, the function f is $(\alpha/\log_2|X|, n(1-1/\alpha), n)$ - independent for $\alpha > 1$.

Corollary 2.1: The sorting function $f_{st}^{(n)}(x_1, \ldots, x_n) = (y_1, \ldots, y_n)$ on n variables where each variable is a binary string over $\{0,1\}^d$, $d \geq \lceil \log_2 n \rceil + 1$, and is lexicographically ordered, is $(\alpha/d^*, n(1-1/\alpha), n)$ - independent for $\alpha > 1$ and $d^* = d - \lceil \log_2 n \rceil$.

Proof

Use the $\lceil \log_2 n \rceil$ most significant bits in each input variable to control an arbitrary permutation of the least significant bits to show that $f_{st}^{(n)}$ is transitive of order n over $\{0,1\}^{d^*}$, $d^* = d - \lceil \log_2 n \rceil$.

This result provides a lower bound on the space-time tradeoff for sorting of the form $(S+1)T \geq (3/32)n^2(d - \lceil \log_2 n \rceil)$ when $\alpha = 2$ and $(S+1)/(d - \lceil \log_2 n \rceil)$ is an integer. This should be compared with the bound of $(S+1)T \geq (15/1024)n^2$ derived by Tompa [2] under the assumption that straight-line algorithms are used with operations consisting of comparisons, selections and Boolean operations. Here S is interpreted as the number of temporary storage registers each capable of containing one word from $\{0,1\}^d$.

Corollary 2.2: The product PAQ of three p×p matrices over the finite field F, of which P and Q are permutation matrices, is $(\alpha/d, p^2(1-1/\alpha), p^2)$ - independent for $d = \log_2|F|$ and $\alpha > 1$. The inverse of a p×p matrix over the finite field F $(\alpha/d, p^2/4(1-1/\alpha), p^2/4)$ - independent for $d = \log_2|F|$ and $\alpha > 1$.

Proof

It is straightforward to show that in PAQ the permutation matrices P and Q induce a group of permutations of components of A which is transitive of order p^2. The result for matrix inversion follows from this and the following identity

$$\begin{bmatrix} P & B \\ 0 & Q \end{bmatrix}^{-1} = \begin{bmatrix} P^{-1} & -P^{-1}BQ^{-1} \\ 0 & Q^{-1} \end{bmatrix}$$

where all matrices are $(p/2)\times(p/2)$, and P and Q are permutation matrices.

From the above it follows that any I/O-oblivious algorithms for matrix inversion over any finite field F requires $(S+1)T \geq (3/32)p^4\log_2|F|$ when $\alpha = 2$ and $(S+1)/\log_2|F|$ is an integer. This should be contrasted with the result of Ja'Ja' [5] for this problem. He considers straight-line algorithms with field operations over the field of rational functions obtained by extending a coefficient field F by the set of indeterminates corresponding to entries in the matrix. Such indeterminates exist only if F is large enough. He then shows by a long proof that the graph of any such straight-line algorithm contains a $p^2/4$-superconcentrator. From this one concludes that $(S+1)T \geq p^2/32$ for S the number of temporary storage registers, using Tompa's lower bound $(S+1)T \geq n^2/2$ for superconcentrators [2]. Thus, our direct approach yields a result weaker by a factor of 3 for a much broader class of algorithms and fields.

We note the fastest known algorithms for matrix inversion use space $S = O(p^2 \log_2|F|)$ and I/O-time $T = O(M(p))$ where $M(p)$ is the number of operations needed to multiply two p×p matrices.

Ja'Ja' observes [5] that any algorithm operating with space S and time T to solve an arbitrary set of p simultaneous linear equations in p unknowns can be used p times to invert a matrix. Thus, $(S+1)T \geq (3/512)p^3 \log_2|F|$ is necessary in our model for this problem.

4. SPACE-TIME TRADEOFFS FOR BANDED MATRICES

A p×p matrix A has a *bandwidth* b if $a_{ij} = 0$ for $|i-j| > b$. In this section we consider the problems of multiplying and inverting banded matrices and of solving sets of simultaneous linear equations whose coefficient matrices are banded. Algorithms as well as lower bounds are considered.

We begin with a few simple observations about banded matrices.

(a) the product of two p×p matrices of bandwidth b is a matrix of bandwidth 2b;

(b) if b is odd and (b+1)/2 divides p, then within every p×p matrix of bandwidth b can be inscribed an m×m block tri-diagonal matrix, $m = p/((b+1)/2)$, in which each block is a $((b+1)/2)\times((b+1)/2)$ matrix *all* of whose components are potentially non-zero. (See the regions enclosed by solid squares in Figure 1.)

(c) If b+1 divides p, then every potentially non-zero entry of a p×p matrix of bandwidth b can be enclosed by a $(p/(b+1))\times(p/(b+1))$ block tri-diagonal matrix whose components are (b+1)×(b+1) matrices. (See the regions enclosed by dashed squares in Figure 1.)

Proposition 3: Let A and B be p×p matrices of bandwidth b over the ring R and let p be divisible by (b+1)/2. Then, the matrix multiplication function defined by $C = A \times B$ is $(\beta, (b+1)/2, d)$ - independent for $\beta = 1/\log_2|R|$ and $d = (3p-b-1)(b+1)/2$. Thus, the space and time required must satisfy

$$\lceil (S+1)/\log_2|R| \rceil T \geq (3/32)(3p-b-1)(b+1)^2.$$

This bound can be achieved up to a constant multiplicative factor.

Proof

In Definition 2 let I_0 consist of the entries in the inscribed block tri-diagonal matrices in A and B, as described in (b) above. Let J_0 be the entries of C in the same positions. Then, $|I_0| = 2|J_0|$, $|J_0| = (3m-2)[(b+1)/2]^2$ and $m = p/((b+1)/2)$. Suppress all entries in C outside of J_0 and zero all entries of A and B outside of $|I_0|$.

For any $k \leq (b+1)/2$, select any k entries of $|I_0|$ and any $((b+1)/2) - k$ entries of $|J_0|$. These latter entries fall into at most a like number of columns of C while the former fall into at most k columns of A. Thus, if B is chosen as a block *diagonal* matrix in which each block is a permutation matrix, it is possible to map columns of A containing no selected entries onto columns of C containing selected entries. It follows that the resulting function has $|R|^{(b+1)/2-k}$ points in the image of its domain, or $\alpha = 1/\log_2|R|$.

The standard straight-line algorithm for this problem performs inner products of rows of A and columns of B. A total of at most n(2b+1) such inner products is computed and each consists of at most 2b+1 multiplications and 2b additions. They can be done in 4 registers each capable of holding

at most one of $|R|$ different values. Thus, $S = O(\log_2|R|)$ and $T = O(nb^2)$, which achieve the lower bound up to a multiplicative constant.

Grigoryev [1] has stated a lower bound of $ST = \Omega(p^3)$ for the multiplication of full p×p matrices over GF(2).

Proposition 4: The function defining the inverse of a p×p matrix of bandwidth b over a finite field F is $(\alpha/d,\ e^2(1-1/\alpha),\ w^2e^2)$ - independent for $\alpha > 1$, $e = (b+1)/2$, $w = \lfloor p/e \rfloor/2 \rfloor$ and $d = \log_2|F|$. Ignoring diophantine constraints, any I/O-oblivious FSA for matrix inversion requires space S and time T satisfying

$$(S+1)T \geq (3/512)p^2(b+1)^2(\log_2|F|)$$

when $\alpha = 2$.

Proof

It is sufficient to consider a banded matrix in which all entries outside the inscribed block tri-diagonal matrix, as described in (b) above, are zero and the block matrices above the diagonal are also zero. If A is the p×p matrix of bandwidth b to be inverted, the resulting m×m block tri-diagonal matrix, $m = \lfloor p/e \rfloor$, $e = (b+1)/2$, has the property that its blocks B_{ij} are zero except for $j = i$, and $j = i-1$ for $i \geq 2$. This matrix B is invertible if and only if $B_{i,i}$ is invertible for $1 \leq i \leq m$. Let C be its inverse, treated as a block matrix. It is straightforward to show that

$$C_{i,j} = \begin{cases} B_{i,i}^{-1} & j=i \\ (-1)^{i+j}B_{i,i}^{-1}\prod_{r=i-1}^{j} B_{r+1,r}B^{r,r-1} & j<i \\ 0 & j>i \end{cases}$$

when B is invertible.

Consider blocks of C defined by $W = \{(i,j) \mid w+1 \leq i \leq m,\ 1 \leq j \leq w\}$ for $w = \lfloor m/2 \rfloor$. Each such block contains the product $B_{w+1,w+1}^{-1}B_{w+1,w}B_{w,w}^{-1}$. Let the diagonal elements of B be the negative inverses of permutation matrices, that is, $B_{i,i} = -P_i^{-1}$. Also, let $B_{i,j} = I$, the e×e identity matrix, for $j = i-1$, except for $i = w+1$, $j = w$. It follows that

$$C_{i,j} = (-1)^{i+j}\left[\prod_{u=w+1}^{i}(-1)P_u\right]B_{w+1,w}\left[\prod_{v=j}^{w}(-1)P_v\right]$$

for $(i,j) \in W$. If in addition $B_{w+1,w} = -H$, then

$$C_{i,j} = U_iHV_j$$

where U_i and V_j are arbitrary permutation matrices and H is an arbitrary matrix over F. By a renumbering of indices, we consider the space and time to compute $C_{i,j} = U_iHV_j$ for $1 \leq i,j \leq w$, $w = \lfloor m/2 \rfloor$.

We now show that the function consisting of the set of products $\{U_iHV_j \mid 1 \leq i,j \leq w\}$ where U_i and V_j are permutation matrices and H is an arbitrary matrix over F is $(\alpha/d,\ e^2(1-1/\alpha),\ w^2e^2)$ - independent for $d = \log_2|F|$.

In Definition 2, let I_0 be the components of the matrix H and let J_0 be the components of the block matrices $C_{i,j}$ for $1 \leq i,j \leq w$. The correspondence with the original elements of A and C is obvious. Select any k elements of I_0 and any l-k elements of J_0. Let

$$a_{r,s} = \begin{cases} 1 & h_{r,s} \text{ selected} \\ 0 & \text{otherwise} \end{cases} \qquad b^{i,j}_{r,s} = \begin{cases} 1 & (C_{i,j})_{r,s} \text{ selected} \\ 0 & \text{otherwise} \end{cases}$$

Now let U_i and V_j be cyclic permutation matrices that provide cyclic shifts by amounts Θ_i and Φ_j, respectively, $0 \le \Theta_i, \Phi_j \le e-1$. If the indices of matrix elements are drawn from the set $\{0,1,2,\ldots,e-1\}$, then the (r,s) component of U_iHV_j is $(C_{i,j})_{r,s} = h_{r+\Theta_i, s+\Phi_j}$ when addition is modulo e. The number of matches between selected inputs and outputs through these permutations is

$$m_{\Theta,\Phi} = \sum_{i,j}\sum_{r,s} b^{i,j}_{r,s} a_{r+\Theta_i, s+\Phi_j}$$

Summing this over all e^{2w} values of $(\Theta,\Phi) = ((\Theta_1, \ldots, \Theta_w),(\Phi_1, \ldots, \Phi_w))$ for $0 \le \Theta_i, \Phi_j \le e-1$ and $1 \le i,j \le w$, we have

$$\sum_{\Theta,\Phi} m_{\Theta,\Phi} = k(l-k)e^{2(w-1)}$$

from which we conclude the existence of Θ_0, Φ_0 such that

$$m_{\Theta_0,\Phi_0} \le (l-k)k/e^2 \le (l-k)l/e^2$$

Consequently, for this assignment (Θ_0,Φ_0) the function has at least $(l-k)(1-l/e^2)$ selected output variables that can be identified with unrestricted input variables. This establishes the desired conclusion.

The following corollary results from the application of an observation of Ja'Ja', as described above.

Corollary 4.1: The function whose value is the solution to a set of p simultaneous linear equations in p unknowns over a finite field F when the coefficient matrix is banded of bandwidth b requires a space-time product which satisfies

$$(S+1)T \ge (3/512)p(b+1)^2(\log_2|F|)$$

when diophantine constraints are ignored.

No algorithms are known which achieve the stated lower bounds within a constant multiplicative factor for banded matrix inversion or for the resolution of a banded set of linear equations. Such a result is not expected for the latter problem because of the difficulty of achieving the lower bound for the non-banded case. We now present some good algorithms for these problems. We begin by presenting an LU decomposition for a banded matrix.

Lemma 1: Let A be an m×m tri-diagonal matrix whose entries are e×e matrices over a finite field F. Then, A can be written as either of two products of upper and lower block diagonal m×m matrices U, L and U^*, L^* where

$$A = LU = U^*L^*$$

and

$$L_{i,j} = \begin{cases} I & j=i \\ a_{i,i-1}x_{i-1}^{-1} & j=i-1 \text{ for } i \ge 2 \\ 0 & \text{otherwise} \end{cases}$$

$$U_{i,j} = \begin{cases} x_i & j=i \\ a_{i,i+1} & j=i+1 \text{ for } i \le m-1 \\ 0 & \text{otherwise} \end{cases}$$

where $x_1 = a_{1,1}$, $x_i = a_{i,i} - a_{i,i-1}x_{i-1}^{-1}a_{i-1,i}$ for $i \ge 2$. Also,

$$L_{i,j}^{*} = \begin{cases} d_i & j=i \\ a_{i,i+1} & j=i-1 \text{ for } i \ge 2 \\ 0 & \text{otherwise} \end{cases}$$

$$U_{i,j}^{*} = \begin{cases} I & j=i \\ a_{i,i+1}d_{i+1}^{-1} & j=i+1 \text{ for } i \le m-1 \\ 0 & \text{otherwise} \end{cases}$$

where $d_m = a_{m,m}$, $d_i = a_{i,i} - a_{i,i+1}d_{i+1}^{-1}a_{i+1,i}$ for $i \le m-1$. The matrix A is non-singular if and only if each of the e×e matrices x_i, d_i, $1 \le i \le m$, is non-singular.

Given a p×p matrix of bandwidth b, inscribe it in and m×m block tri-diagonal matrix A of e×e matrices, $m = p/e$, $e = b+1$, as described in (c) at the beginning of this section. We now show that standard Gaussian elimination with forward elimination and backward substitution on the block matrix can be reduced to computations for which efficient space-limited algorithms are known.

The object is to solve the set of equations

$$\mathbf{Az} = \boldsymbol{\beta} \tag{1}$$

for A as described above and $\mathbf{z}$ and $\boldsymbol{\beta}$ two "block" m×1 matrices whose components are e×1 matrices over F. Let these "blocks" be $\{z_i, \beta_i \mid 1 \le i \le m\}$. The algorithms to be developed can be used with the $p = me$ unit vectors to invert the matrix in question. Let $\mathbf{y}$ satisfy

$$\mathbf{Uz} = \mathbf{y} \tag{2}$$

Then, to solve (1) we first solve

$$\mathbf{Ly} = \boldsymbol{\beta} \tag{3}$$

and then solve (2). The solutions to (3) are

$$y_1 = \beta_1, \quad y_i = \beta_i - a_{i,i-1}x_{i-1}^{-1}y_{i-1} \text{ for } i \ge 2. \tag{4}$$

and the solutions to (2) are

$$z_m = x_m^{-1}y_m, \quad z_j = x_m^{-1}(y_j - a_{j,j+1}z_{j+1}) \text{ for } j \le m-1. \tag{5}$$

Combine the computation of x_j^{-1} with that of y_i in one step s_i. Then, this step requires one matrix inversion at a cost of I(e), two matrix multiplications at a cost of M(e) each, one matrix-vector multiplication at a cost of $2e^2$ operations, and one matrix subtraction at a cost of e^2 each, for a total incremental cost Δs_i of at most

$$\Delta s_i = I(e) + M(e) + 5e^2 \tag{6}$$

Each z_j can be computed in a step r_j at an incremental cost of at most Δr_j where

$$\Delta r_j = 5e^2 \tag{7}$$

Since matrix inversions are no more costly that some constant multiple of M(e) [11], it follows that both costs are O(M(e)). In both cases, temporary

storage is needed for two exe matrices and for two ex1 vectors to do the incremental computations.

The graph of the dependence between steps $\{s_i, r_j\}$ is a ***ladder network***, as is shown in Figure 2. Note that reads of inputs β_i, $a_{i,i-1}$, and $a_{i-1,i}$ are not shown explicitly but are subsumed in the individual steps. The ladder network corresponds to a computation on a stack in which the results of computation steps $s_1, s_2, \ldots, s_m$ are pushed onto a stack in this order and popped in reverse order to compute the results $r_m, r_{m-1}, \ldots, r_1$. If insufficient storage is available to hold all the entries in a stack, it is possible to save certain positions in a stack as starting points to recompute virtual stack elements. Of course, this recomputation requires time and the possible trades of stack space for time are of interest.

Chandra [12] and Swamy and Savage [13] have considered space-time tradeoffs for the graph of Figure 2. Space is modeled with "pebbles". A pebble is placed on a node to signify that the results of the corresponding computations are held in a register. A pebble can be placed on an empty node only if all nodes with edges directed into it contain pebbles. It is possible in this case to cover the empty node with a pebble from one of its predecessors, which corresponds to re-using a register. The graph in question, the ladder network, can be pebbled with just two pebbles, one for the upper spine and one for the lower. However, when just two pebbles are used, the number of moves required is proportional to m^2. This yields a space-time product of $ST = O(p^2M(e)\log_2|F|)$ which is much larger than the lower bound given in Corollary 4.1. Swamy and Savage [13] determine the optimal placement of a given number of ***pebbles*** p, on the ladder network of ***depth*** n sot that the number of ***moves*** M(n,p) is minimized. For large n, they show that

$$M(n,p) = \begin{cases} O(pn^{1+1/p}) & p = o(\log n) \\ O(n\log n/\log p) & \omega(\log n) \\ O(n\log n) & \Theta(\log n) \end{cases}$$

Thus, if $p = \sqrt{n}$, which represents a dramatic reduction in space, the number of moves remains proportional to n.

The value of p that minimizes the space-time product is $p = O(\log n)$. This implies that space for $\Theta(\log n)$ exe matrices is used and on the order of nlog n node computations are performed, each at a cost of O(M(e)). This yields the following result.

Proposition 5: The solution of a banded set of p linear equations in p unknowns with bandwidth b over a finite field F can be obtained as a pebbling of a ladder network of depth $p/(b+1)$. A space-time product of

$$(S+1)T = O(pbM(b)(\log p/b)^2\log|F|)$$

can be achieved. Inversion of a banded matrix can be done with the following space-time product:

$$(S+1)T = O(p^2M(b)(\log p/b)^2\log|F|)$$

Proof

The upper bound for the problem of resolving a set of equations follows from the above discussion.

The upper bound for matrix inversion is obtained by noting that the algorithm for the above problem can be used p times with the p unit vectors.

This, however, leads to a bound which is larger than the quoted result by a factor of b. The elimination of this factor is obtained by noting that if we solve e sets of equations simultaneously, and in parallel, the cost of steps s_i and r_j, as stated in (5) and (6), increase to $I(e) + 4M(e) + e^2$ and $2M(e) + e^2$, respectively, when e matrix-vector multiplications are done as e×e matrix multiplications. Storage space increases to space for the e×e matrices from that for one such matrix and two e×1 vectors. Thus, the space-time product is on the order of p/e times larger than that for the solution of a set of equations.

The upper bound for matrix inversion demonstrates that the lower bound of Proposition 4 is smaller by two factors, $M(b)/b^2$, which reflects our ignorance of the cost of matrix multiplication, and $(\log p/b)^2$. The latter term reflects the recursive nature of our pebbling algorithm. If $b = O(\sqrt{p})$, as it is for many problems, such as for the solution by finite differences of Laplace's equation, this term is much smaller than b^2. We note that if a given problem, as represented by a banded p×p coefficient matrix A, is to be solved $\Theta(b)$ times, it is more efficient to combine operations in (5) and (6) above than to solve $\Theta(b)$ individual problems. We also note that in the limit of b large, these bounds reduce to those for the general case.

Before closing we return to the result of Eisenstat et al [7] that motivated the author's interest in banded matrix problems. They present an algorithm for the solution of a banded set of equations whose matrix is positive definite and which uses time $O(pb^2(\log p/b))$ and space $O(b^2(\log p/b))$. (The latter factor of $(\log p/b)$ was not reported in their paper and is necessary to implement their recursive algorithms.) We give a block version of their algorithm.

Represent the matrix A as suggested in (c) at the beginning of this section. Let $e = b+1$ divide p and let $m = p/e$. From Lemma 1 we can write

$$\mathbf{Az} = \mathbf{LUz} = \mathbf{U^*L^*z} = \boldsymbol{\beta} \tag{8}$$

and

$$\mathbf{Uz} = \mathbf{r}, \quad \mathbf{Lr} = \boldsymbol{\beta} \tag{9}$$

$$\mathbf{L^*z} = \mathbf{s}, \quad \mathbf{U^*s} = \boldsymbol{\beta} \tag{10}$$

From these equations we can solve for r_k, x_k and s_{k+1}, d_{k+1} using the following identities:

$$r_1 = \beta_1, \; r_i = \beta_i - a_{i,i-1}x_{i-1}^{-1}r_{i-1}, \quad 2 \le i \le m$$

$$x_1 = a_{1,1}, \; x_i = a_{1,1} - a_{i,i-1}x_{i-1}^{-1}a_{i-1,i}, \quad 2 \le i \le m$$

$$s_m = \beta_m, \; s_j = \beta_j - a_{j,j+1}d_{j+1}^{-1}s_{j+1}, \quad 1 \le j \le m-2$$

$$d_m = a_{m,m}, \; d_j = a_{j,j} - a_{j,j+1}d_{j+1}^{-1}a_{j+1,j}, \quad 1 \le j \le m-2$$

using $O(M(e)m)$ operations over F and space to hold $O(e^2)$ elements from F. Then using (9) we have

$$\begin{bmatrix} x_k & a_{k,k+1} \\ a_{k+1,k} & d_{k+1} \end{bmatrix} \begin{bmatrix} z_k \\ z_{k+1} \end{bmatrix} = \begin{bmatrix} r_k \\ s_{k+1} \end{bmatrix} \tag{11}$$

which can be solved for z_k and z_{k+1} in $O(M(e))$ time and $O(e^2)$ space. Substitute z_k and z_{k+1} back into (8) to divide the original problem into two problems corresponding to block matrices which are $(k-1)\times(k-1)$ and $(m-k-1)\times(m-k-1)$.

Thus, the problem can be solved by recursion. If we choose $k = \lceil m/2 \rceil$, the time involved satisfies

$$T(m) \leq T(\lceil m/2 \rceil - 1) + T(m - \lceil m/2 \rceil - 1) + O(M(e)m)$$

which satisfies

$$T(m) \leq O(M(e)m \log m)$$

The space required is $O((e^2 \log |F|) \log m)$ since space of $O(e^2 \log |F|)$ is required for each of the $O(\log m)$ levels of recursion. Thus, a space-time product of the order as indicated in Proposition 5 is used. The reduction of the time by a factor of $M(e)/e^3$ by comparison with the Eisenstat et al result is possible because of the treatment of A as a block tri-diagonal matrix.

5. CONCLUSIONS

In this paper we have extended the conditions under which the Grigoryev method for the study of space-time tradeoffs can be applied and we have used this method to derive lower bounds for several problems. A lower bound for the shifting function implies a lower bound for certain binary integer functions including integer division and square roots. A lower bound for the class of transitive functions implies lower bounds for sorting and for matrix inversion over an arbitrary finite field. Problems having to do with banded matrices, however, are the principal topics treated in this paper. Lower bounds have been derived for the multiplication and the inverse of banded matrices and for solving a banded set of linear equations. Upper bounds for the banded multiplication problem match the lower bounds up to a multiplicative constant. The upper bounds derived for the other two problems are less good, with the bound for matrix inversion being closer to the lower bound than that for resolving a set of equations.

The lower bound on space time exchanges for resolving a full set of linear equations in p unknowns over a finite field F is

$$(S+1)T \geq (3/512)p^3 \log_2 |F|$$

when diophantine constraints are ignored. The best known upper bound, however, is

$$(S+1)T \leq O(M(p)p^2 \log_2 |F|)$$

which is achieved by the LUP method outlined in [11, p.233]. There is an important gap here that needs to be closed. The lower bound for matrix inversion is p times as large as the lower bound for resolving a set of equations. The best known upper bound for inversion is that given above for the problem of resolving a set of linear equations. Thus, the ratio of the two bounds in this case is $M(p)/p^2$, where $M(p)$ is the number of operations to multiply two $p \times p$ matrices. This ratio may in fact be a constant.

The lower bound for the resolution of a banded set of linear equations with bandwidth b is

$$(S+1)T \geq (3/512)p(b+1)^2(\log_2 |F|)$$

while the best known upper bound is

$$(S+1)T = O(pbM(b)(\log p/b)^2 \log |F|)$$

The gap is measured here by a factor of $O(M(b)/b)$ which is at least linear in b. The upper bound for banded matrix inversion is

$$(S+1)T = O(p^2M(b)(\log p/b)^2 \log |F|)$$

and the gap here is measured by a factor of $O(M(b)/b^2)$, which may in fact be a constant, and a factor of $(\log p/b)^2$, which can be small if the ratio p/b is not too large. It remains to be seen if the lower bound for the resolution of a set of banded linear equations is tight or not.

The upper bound derived by Eisenstat et al [7] has been improved. Theirs was originally derived under the assumption that the banded matrices were positive definite. This condition is not used here in the derivation of upper or lower bounds. It is of interest to extend the upper and lower bounds to this case. It is also of interest to treat the large class of special problems that are often encountered in practice.

6. REFERENCES

1. D. Yu. Grigoryev, "An Application of Separability and Independence Notions for Proving Lower Bounds of Circuit Complexity," *Notes of Scientific Seminars, Steklov Math. Inst.* **60** pp. 35-48 (1976).
2. M. Tompa, "Time-Space Tradeoffs for Computing Functions, Using Connectivity Properties of Their Circuits," *Proc. 10th Ann. ACM Symp. Th. Comp.*, pp. 196-204 (May 1978).
3. Sowmitri Swamy, "On Space-Time Tradeoffs," Ph.D. Thesis, Division of Engineering, Brown University, Providence, RI (June 1978).
4. J. E. Savage and S. Swamy, "Space-Time Tradeoffs for Oblivious Integer Multiplication," pp. 498-504 in *Lecture Notes in Computer Science*, ed. H. A. Maurer, Springer-Verlag, Berlin, Heidelberg, New York (July, 1979).
5. J. Ja'Ja', "Time-Space Tradeoffs for Some Algebraic Problems," *12th Ann. ACM Symp. on Th. Computing*, pp. 339-350 (April 28-30, 1980).
6. O. C. Zienkiewicz, *The Finite Element Method*, McGraw-Hill, London (1977).
7. S. C. Eisenstat, M. H. Schultz, and A. H. Sherman, "Minimal Storage Band Elimination," pp. 273-286 in *High-Speed Comp. and Algorithm Org.*, ed. D. J. Kuck, D. H. Lawrie, A. H. Sameh, Academic Press (1977).
8. L. G. Valiant, *Personal Communication*
9. J. Vuillemin, "A Combinatorial Limit to the Computing Power of VLSI Circuits," *Procs. 21st Ann. Symp. Th. Comp.*, pp. 294-300 (Oct. 12-14, 1980).
10. J. E. Savage, "Planar Circuit Complexity and the Performance of VLSI Algorithms," Report No. 69, Dept. of Comp. Sci., Brown U., Providence, RI (April 1981). Also appears as INRIA report No. 77
11. A. V. Aho, J. E. Hopcroft, and J. D. Ullman, *The Design and Analysis of Computer Algorithms*, Addison Wesley (1974).
12. A. K. Chandra, "Efficient Compilation of Linear Recursive Programs," *Proc. 14th Ann. Symp. on Switching and Aut. Th.*, pp. 16-25 (Oct. 15-17, 1973).
13. J. E. Savage and S. Swamy, "Space-time Tradeoffs for Linear Recursion," *The 6th Ann. ACM Symp. Princ. Prog. Langs.*, pp. 135-142 (Jan. 1979).

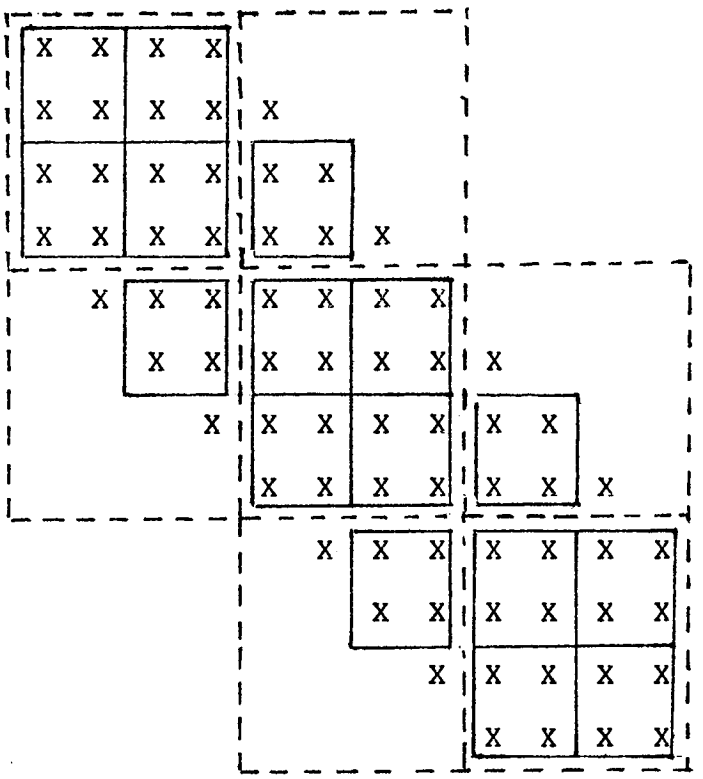

Figure 1. A Banded Matrix with Inscribed and Superscribed Square Matrices

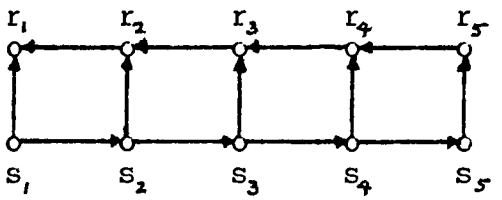

Figure 2. Ladder Network